AMENDMENTS TO THE CLAIMS

1. (Previously Presented) A data processing method including receiving input data containing a plurality of instruction codes, and judging whether or not a process executed based on the instruction codes contained in the received data is a malicious process, said method comprising:

sequentially reading one byte of the input data at a time;

determining whether or not the read data is a branch instruction;

if the read input data is a branch instruction, determining whether a branch destination address of the branch instruction is larger than a branch origin address based only on the one byte of the data read and if the branch destination address is larger than the branch origin address storing the branch destination address and branch origin address;

determining whether or not there is a call instruction at the branch destination address, and storing a call destination address of the call instruction if the instruction code at the branch destination address is a call instruction;

determining whether or not the stored call destination address is between the branch origin address and the branch destination address; and

if the stored call destination address is between the branch origin address and the branch destination address concluding that the input data includes a malicious process.

2. (Previously Presented) A data processor including means for receiving input data containing a plurality of instruction codes, for determining whether or not a process executed

based on the instruction codes contained in the received data is a malicious process, said data processor comprising:

means for sequentially reading one byte of the input data at a time;

means for determining whether the read data is a branch instruction;

if the read input data is a branch instruction determining whether a branch destination address of the branch instruction is larger than a branch origin address based only on the one byte of the data read and if the branch destination address is larger than the branch origin address storing the branch destination address and the branch origin address;

means for determining whether or not there is a call instruction at the branch destination address, and storing a call destination address of the call instruction if the instruction code at the branch destination address is a call instruction; and

means for determining whether or not the stored call destination address is between the branch origin address and the branch destination address; and

if the stored call destination address is between the branch origin address and the branch destination address concluding that the input data includes a malicious process.

3. (Currently Amended) The data processor as set forth in claim 2, further comprising means for judging whether or not a predetermined character string is associated with a return address of ~~the~~ an instruction code group called by the call instruction, wherein if the predetermined character string is associated with the return address, the information indicating that the data is data for executing a malicious process is outputted.

4- 6. (Canceled)

7. (Previously Presented) A computer-readable memory product storing a computer program including causing a computer to judge whether or not a process executed based on input data containing a plurality of instruction codes is a malicious process, the stored computer program comprising:

causing the computer to sequentially read one byte of the input data at a time;

causing the computer to determine whether or not the read data is a branch instruction;

if the read input data is branch instruction, causing the computer to determine whether a branch destination address of the branch instruction is larger than a branch origin address based only on the one byte of the data read, and if the branch destination address is larger than the branch origin address causing the computer to store the branch destination address and branch origin address;

causing the computer to determine whether or not there is a call instruction at the branch destination address and to store a call destination address of the call instruction if the instruction code at the branch destination address is a call instruction;

causing the computer to determine whether or not the stored call destination address is between the branch origin address and the branch destination address; and

if the stored call destination address is between the branch origin address and the branch destination address causing the computer to conclude that the input data includes a malicious process.

8. (Previously Presented) A data processor comprising:

an input unit for inputting data containing a plurality of instruction codes;

a storing unit for storing the data input by the input unit; and

a controller capable of performing operations of;

sequentially reading one byte of the input data at a time;

determining whether or not the read data is a branch instruction;

if the read data is a branch instruction determining whether a branch destination address of the branch instruction is larger than a branch origin address based only on the one byte of the data read, and if the branch destination address is larger than the branch origin address storing the branch destination address and branch origin address;

determining whether or not there is a call instruction at the branch destination address and storing a call destination address of the call instruction in the storing unit if the instruction code at the branch destination address is a call instruction;

determining whether or not the stored call destination address is between the branch origin address and the branch destination address; and

if the stored call destination address is between the branch origin address and the branch destination address concluding that the input data includes a malicious process.

9-10. (Canceled)

11. (Previously Presented) The data processing method according to claim 1, wherein the malicious process causes an erroneous operation in the process executed based on the instruction codes contained in the received data.